

EXHIBIT 031


U.S. Patent No. 8,086,800 (Radulescu and Goossens)*“Integrated circuit and method for buffering to optimize burst length in networks on chips”*

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
<p>10. A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions, the method comprising the acts of:</p>	<p>Without conceding that the preamble of claim 10 of the '800 Patent is limiting, the Lenovo IdeaPad Duet 3 Chromebook (hereinafter, the “Lenovo product”) performs a method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions), either literally or under the doctrine of equivalents.</p> <p>The Lenovo product includes an integrated circuit. For example, the Lenovo product includes the Qualcomm Snapdragon 7c Gen 2 Compute Platform system on chip (hereinafter, the “Snapdragon SoC”).</p>

¹ The Lenovo product is charted as a representative product made used, sold, offered for sale, and/or imported by Lenovo. The citations to evidence contained herein are illustrative and should not be understood to be limiting. The right is expressly reserved to rely upon additional or different evidence, or to rely on additional citations to the evidence cited already cited herein.


U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div data-bbox="499 310 1010 678">  </div> <div data-bbox="1115 264 1696 378"> <h2>Lenovo IdeaPad Duet 3 Chromebook</h2> </div> <div data-bbox="1115 402 1696 430"> <p>Featuring a Snapdragon 7c Gen 2 Compute Platform</p> </div> <div data-bbox="1115 451 1793 727"> <p>The Lenovo IdeaPad™ Duet 3 Chromebook is the ideal work and play device for the hyper-mobile user looking for superior experience with the larger 11" 2K near-borderless display. Faster connectivity options, all-day battery life, and the more powerful, fanless and efficient performance of the Snapdragon® 7c Gen 2 platform gets things done while on the go. Work on the detachable keyboard or take notes and sketch with the optional Lenovo USI Pen 2.</p> </div> <div data-bbox="632 768 877 829"> <p>1 2 3 4</p> </div> <div data-bbox="1121 802 1289 857"> <p>Learn More</p> </div> <div data-bbox="499 889 1751 971"> <p>https://www.qualcomm.com/products/application/mobile-computing/laptop-device-finder/lenovo-ideapad-duet-3-chromebook</p> </div> <div data-bbox="499 1008 1873 1084"> <p>The Snapdragon SoC comprises a plurality of processing modules, for example Qualcomm Adreno GPU; Octa-core Qualcomm Kryo 468 CPU; and Qualcomm Hexagon 692 DSP:</p> </div>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div data-bbox="548 250 1822 354"> <p>Qualcomm® Snapdragon™</p> <p>7c Gen 2 Compute Platform</p> <p>Qualcomm snapdragon </p> </div> <div data-bbox="548 472 911 508">Specifications & Features</div> <div data-bbox="548 540 596 565">CPU</div> <div data-bbox="548 581 911 683"> <ul style="list-style-type: none"> • CPU Clock Speed: Up to 2.55 GHz • CPU Cores: Octa-core Qualcomm® Kryo™ 468 CPU • CPU Architecture: 64-bit </div> <div data-bbox="548 704 625 729">Process</div> <div data-bbox="548 745 789 769"> <ul style="list-style-type: none"> • Process Technology: 8 nm </div> <div data-bbox="548 789 661 813">OS Support</div> <div data-bbox="548 829 884 878"> <ul style="list-style-type: none"> • Supports Windows 10 and Windows 11 • Chrome OS </div> <div data-bbox="548 899 632 924">Memory</div> <div data-bbox="548 940 900 964"> <ul style="list-style-type: none"> • Memory Type: 2 x 16-bit, LPDDR4x-4266 </div> <div data-bbox="548 984 630 1008">Storage</div> <div data-bbox="548 1024 768 1049"> <ul style="list-style-type: none"> • UFS: eMMC 5.1; UFS 2.1 </div> <div data-bbox="548 1068 716 1092">Visual Subsystem</div> <div data-bbox="548 1109 837 1133"> <ul style="list-style-type: none"> • GPU: Qualcomm® Adreno™ GPU </div> <div data-bbox="548 1153 630 1177">Camera</div> <div data-bbox="548 1193 926 1263"> <ul style="list-style-type: none"> • Image Signal Processor: Qualcomm Spectra™ 255 image signal processor, 14-bit • Dual Camera, ZSL, 30fps: Up to 16 MP </div> <div data-bbox="984 540 1045 565">Video</div> <div data-bbox="984 581 1373 708"> <ul style="list-style-type: none"> • Video Playback: Up to 4K HDR10 • Codec Support: H.265 (HEVC), H.264 (AVC), VP9 • Video Software: Motion Compensated Temporal Filtering (MCTF) </div> <div data-bbox="984 729 1060 753">Display</div> <div data-bbox="984 769 1337 870"> <ul style="list-style-type: none"> • Max On-Device Display: QXGA @ 60Hz, FHD @ 60Hz • Max External Display: QHD @ 60Hz • Display Pixels: 2560x1440, 2048x1536 </div> <div data-bbox="984 893 1121 915">General Audio</div> <div data-bbox="984 932 1362 1052"> <ul style="list-style-type: none"> • Qualcomm Aqstic technology: Qualcomm Aqstic™ audio codec, Qualcomm Aqstic smart speaker amplifier • Qualcomm® aptX™ audio playback support: aptX, aptX HD </div> <div data-bbox="984 1073 1134 1097">Audio Playback</div> <div data-bbox="984 1114 1352 1185"> <ul style="list-style-type: none"> • PCM, Playback: Up to 384kHz/32bit • Additional Playback Features: Native DSD support </div> <div data-bbox="984 1205 1190 1229">Qualcomm® AI Engine</div> <div data-bbox="984 1245 1295 1269"> <ul style="list-style-type: none"> • AIE CPU: Octa-core Kryo 468 CPU </div> <div data-bbox="1415 540 1814 729"> <ul style="list-style-type: none"> • Uplink Technology: Qualcomm® Snapdragon™ Upload+ • Uplink Carrier Aggregation: 2x20 MHz carrier aggregation • Uplink QAM: Up to 64-QAM • LTE Speed • LTE Peak Download Speed: 600 Mbps </div> <div data-bbox="1415 750 1478 774">Wi-Fi</div> <div data-bbox="1415 789 1761 924"> <ul style="list-style-type: none"> • Wi-Fi Standards: 802.11ac Wave 2, 802.11a/b/g, 802.11n • Wi-Fi Spectral Bands: 24 GHz, 5 GHz • MIMO Configuration: 2x2 (2-stream) • Qualcomm® FastConnect™ Subsystem </div> <div data-bbox="1415 943 1589 967">Bluetooth Version</div> <div data-bbox="1415 984 1560 1008"> <ul style="list-style-type: none"> • Bluetooth 5.0 </div> <div data-bbox="1415 1027 1554 1052">GPS Location</div> <div data-bbox="1415 1068 1793 1114"> <ul style="list-style-type: none"> • Satellite Systems Support: NavIC, BeiDou, Galileo, GLONASS, GPS, QZSS, SBAS </div> <div data-bbox="1415 1131 1503 1156">Security</div> <div data-bbox="1415 1172 1707 1252"> <ul style="list-style-type: none"> • Qualcomm® Processor Security • Qualcomm® Content Protection • Wi-Fi Security: WPA3 </div>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div data-bbox="506 250 1885 857"> <p>Camera</p> <ul style="list-style-type: none"> Image Signal Processor: Qualcomm Spectra™ 255 image signal processor, 14-bit Dual Camera, ZSL, 30fps: Up to 16 MP Single Camera, ZSL, 30fps: Up to 32 MP Camera Features: Multi-frame Noise Reduction (MFNR) Video Capture Features: Rec. 2020 color gamut video capture, Up to 10-bit color depth video capture <p>CAMERA FEATURES</p> <ul style="list-style-type: none"> Advanced DPD, WPA3 Multi-Frame Noise Reduction (MFNR) and Multi-Frame Super Resolution (MFSR) Forward-looking Electronic Image Stabilization (EIS) Motion Compensated Temporal filtering (MCTF) for noise-free video capture up to UHD (4K) at 30 FPS Four MIPI CSI PHYs (DPHY 1.2 / CPHY 1.2) <p>Qualcomm® AI Engine</p> <ul style="list-style-type: none"> AI Engine CPU: Octa-core Kryo 468 CPU AI Engine GPU: Adreno GPU AI Engine DSP: Qualcomm® Hexagon™ 692 DSP <p>Cellular Modem</p> <ul style="list-style-type: none"> Modem Name: Snapdragon X15 LTE modem LTE Category Downlink LTE Category: LTE Category 12 Uplink LTE Category: LTE Category 13 LTE Downlink Features Downlink Carrier Aggregation: 3x20 MHz carrier aggregation Downlink LTE MIMO: Up to 4x4 MIMO on two carriers Downlink QAM: Up to 256-QAM, Up to 64-QAM LTE Uplink Features <p>Additional Playback Features: Native DSD support</p> <ul style="list-style-type: none"> Qualcomm® Processor Security Qualcomm® Content Protection Wi-Fi Security: WPA3 </div> <p>https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/prod_brief_qcom_sd7c_gen2.pdf</p> <p>The Snapdragon SoC included in the Lenovo product utilizes Arteris network on chip interconnect technology, and/or a derivative thereof, (collectively, the “Arteris NoC”) as an interconnect to connect the plurality of processing modules through interface units:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div data-bbox="512 256 1066 940"><p data-bbox="558 305 768 354">Qualcomm</p><p data-bbox="558 557 1003 735">Arteris-developed NoC technology is the backbone of Snapdragon application processors & LTE modems, Atheros wireless connectivity SoCs, and CSR IoT products.</p><div data-bbox="661 805 909 878">LEARN MORE »</div></div> <p data-bbox="501 992 1713 1027">https://web.archive.org/web/20210514110614/https://www.arteris.com/customers</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="575 253 1360 302">Certain Arteris Technology Assets Acquired</p> <p data-bbox="785 331 1150 358">by Kurt Shuler, on October 31, 2013</p> <p data-bbox="512 396 1255 423">Arteris to continue to license, support and maintain Arteris FlexNoC® interconnect IP</p> <p data-bbox="512 448 1417 553">SUNNYVALE, California — October 31, 2013 — Arteris Inc., a leading innovator and supplier of silicon-proven commercial network-on-chip (NoC) interconnect IP solutions, today announced that Qualcomm Technologies, Inc. (“Qualcomm”), a subsidiary of Qualcomm Incorporated, has acquired certain technology assets from Arteris and hired personnel formerly employed by Arteris.</p> <p data-bbox="512 586 1360 732"> “Arteris NoC technology has been and will continue to be a key enabler for creating larger and more complex chips in a shorter amount of time at a lower cost. This acquisition of our technology assets represents a validation of the value of Arteris’ Network-on-Chip interconnect IP technology. ” </p> <p data-bbox="1234 776 1377 803">ARTERIS IP</p> <p data-bbox="1119 850 1377 868"><i>K. Charles Janac, President and CEO, Arteris</i></p> <p data-bbox="504 935 1797 1008"> https://www.arteris.com/press-releases/Qualcomm-Arteris-asset-acquisition-2013_oct_31; https://www.fiercewireless.com/tech/qualcomm-acquires-arteris-noc-tech-assets-team </p> <p data-bbox="504 1052 1850 1125">A large SoC, such as the Snapdragon SoC included in the Lenovo product may include multiple classes of Arteris NoC interconnect:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div data-bbox="533 261 1577 315"> <h2>Logical Interconnect Topology Development</h2> </div> <div data-bbox="533 326 1402 354"> <p>FLEXNOC & NCORE INTERCONNECT IPS DEFINE ARCHITECTURES</p> </div> <div data-bbox="533 370 1864 805"> </div> <div data-bbox="533 813 1745 915"> <ul style="list-style-type: none"> • ArChip16 Example: Large SoCs have multiple classes of interconnect <ul style="list-style-type: none"> – Non-coherent, Coherent, Control/Status, Observability, etc. • Ncore & FlexNoC interconnects are managed separately from IP blocks, increasing design flexibility </div> <div data-bbox="504 954 638 979"> <p>ARTERIS IP</p> </div> <div data-bbox="1098 958 1253 974"> <p>ISPD 2018, 28 March 2018</p> </div> <div data-bbox="1638 958 1848 974"> <p>Copyright © 2018 Arteris IP 9</p> </div>
	<p>See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 9.</p> <p>The Arteris NoC is an interconnect connects the plurality of processing modules in the Snapdragon SoC included in the Lenovo product through interface units, wherein a first processing module communicates to a second processing module using transactions.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris</p>

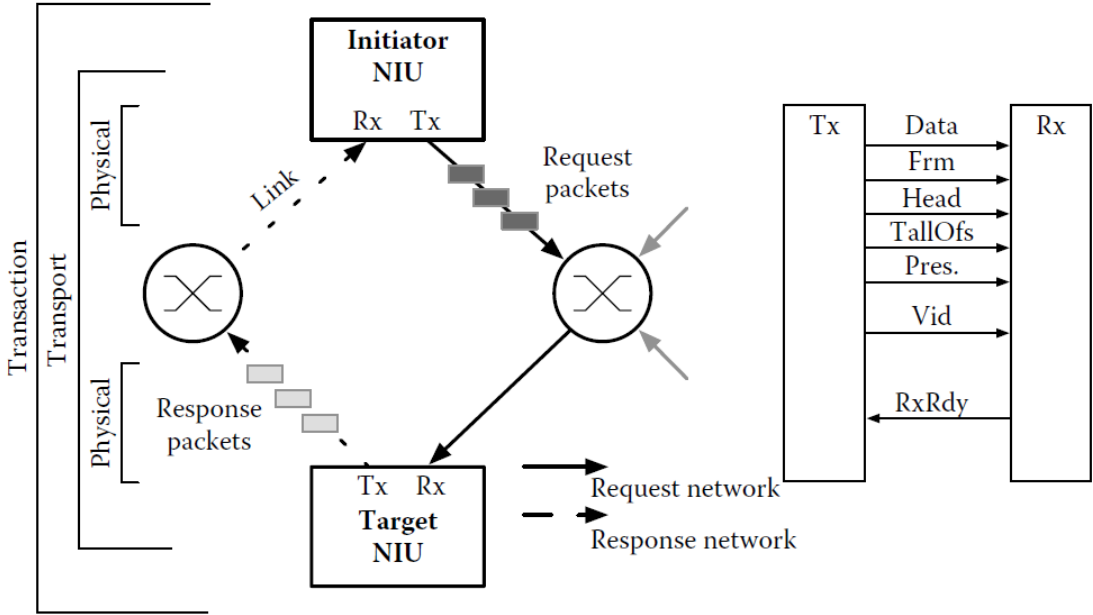
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>NoC, “[m]ost transactions require the following two-step transfers,” including “[a] master send[ing] request packets” and “the slave return[ing] response packets”:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p> <p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU’s Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313; see <i>id</i> at 308 (explaining that Chapter 11 of this book describes the function of the Arteris NoC: “In this chapter we will present an MPSoC platform [...] using Arteris NoC as communication infrastructure.”).</p>
configuring the first processing module having a first memory as a	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product configures the first processing module having a first memory as a master the provides requests, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
master the provides requests;	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, “[m]ost transactions require the following two-step transfers,” including “[a] master send[ing] request packets” and “the slave return[ing] response packets”:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

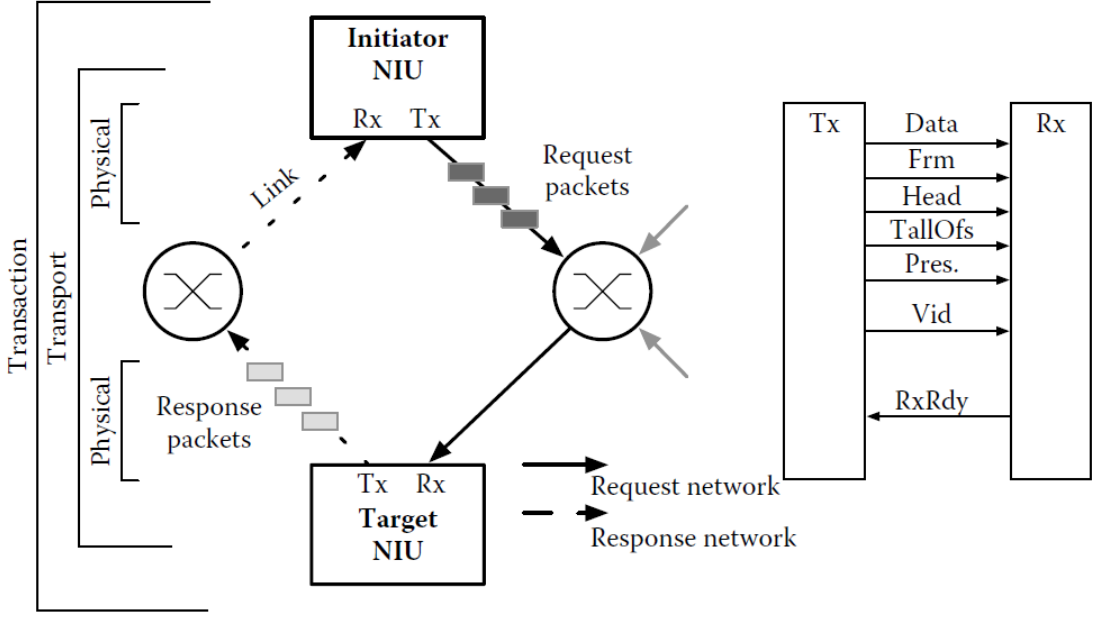
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 732 1808 805">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1887 1000">As a further example, “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC [and] translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP” and has a “FIFO memory [...] inserted in the datapath for AHB write access”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.1 Initiator NIU Units</p> <p>Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p> <p>The diagram illustrates the NIU Architecture, which is divided into two main sections: the Request Path and the Response Path. Both paths interface with an AHB Slave Interface on the left. The Request Path starts with an AHB Req (indicated by a thick arrow) entering the AHB Slave Interface. Data flows from the AHB Slave Interface into a DATA FIFO, then to a Packet Assembly block, and finally to a PIPE bw/lw block, which connects to the Tx Port. The Response Path starts with an AHB Resp (indicated by a thick arrow) leaving the AHB Slave Interface. It flows through a FLOW CONTROL block, then to a PIPE block, and finally to a WIDTH CONVERTER (dashed box), which connects to the Rx Port. A TRANSLATION TABLE block is connected to the AHB Slave Interface and the Request Path. A BUILD HEADER & NECKER block is connected to the TRANSLATION TABLE and the Packet Assembly. A CONTROL line connects the FLOW CONTROL block to the PIPE bw/lw block. Information from the request path is also fed back into the FLOW CONTROL block.</p> <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p>
configuring the second processing module having a second memory as	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product configures the second processing module having a second memory as a slave the provides responses to the requests, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
a slave the provides responses to the requests;	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, “[m]ost transactions require the following two-step transfers,” including “[a] master send[ing] request packets” and “the slave return[ing] response packets”:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

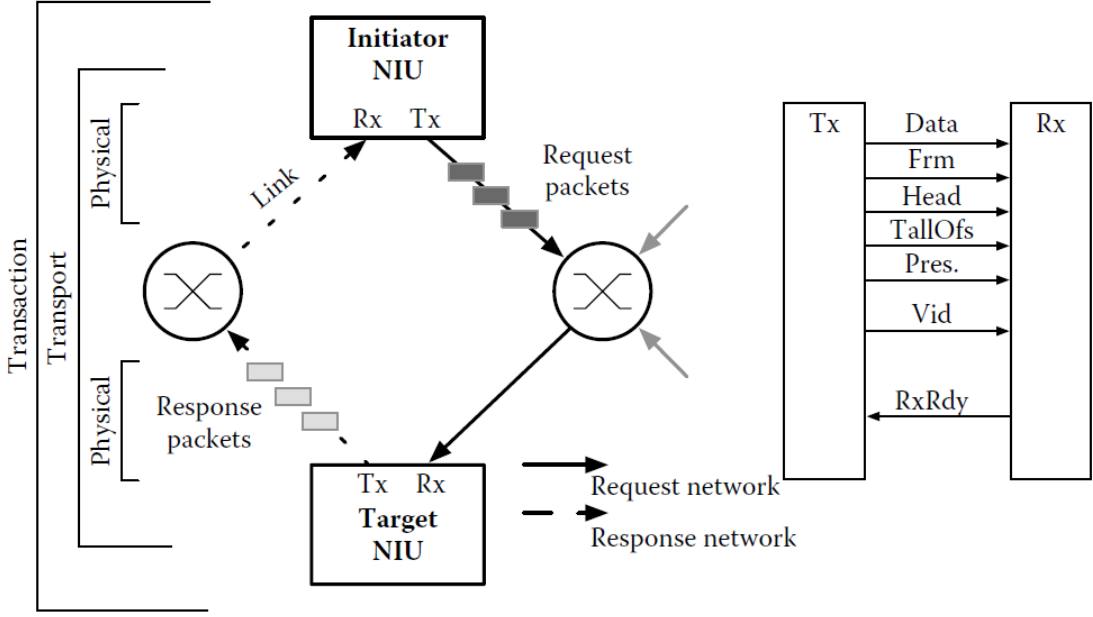
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2 Network Interface Units</p> <p>The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p>As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets” and have a FIFO memory in the datapath:</p>

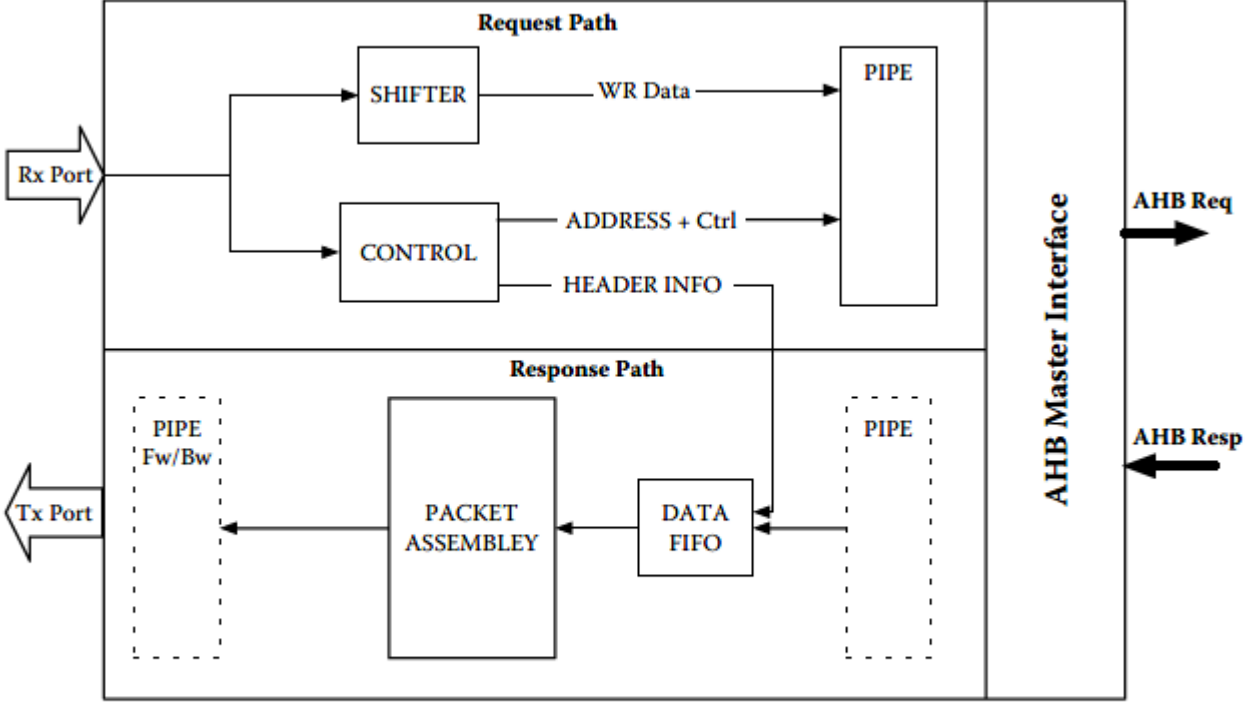
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">Target NIU Architecture</p>  <p>The diagram illustrates the Target NIU Architecture. It is divided into two main horizontal sections: the Request Path (top) and the Response Path (bottom). On the left, there is an Rx Port (input) and a Tx Port (output). The Request Path contains a SHIFTER and a CONTROL block. The SHIFTER outputs WR Data to a PIPE. The CONTROL block outputs ADDRESS + Ctrl and HEADER INFO to the same PIPE. The Response Path contains a DATA FIFO, a PACKET ASSEMBLY, and two dashed boxes labeled PIPE Fw/Bw. The DATA FIFO outputs to the PACKET ASSEMBLY, which then outputs to the left PIPE Fw/Bw. The PACKET ASSEMBLY also outputs to the right PIPE Fw/Bw. The HEADER INFO from the Request Path is fed into the DATA FIFO. On the right, the AHB Master Interface connects to the system, with AHB Req (request) and AHB Resp (response) signals.</p> <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>
connecting the master to a master	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product connects the master to a master interface unit of the interface units, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
interface unit of the interface units;	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

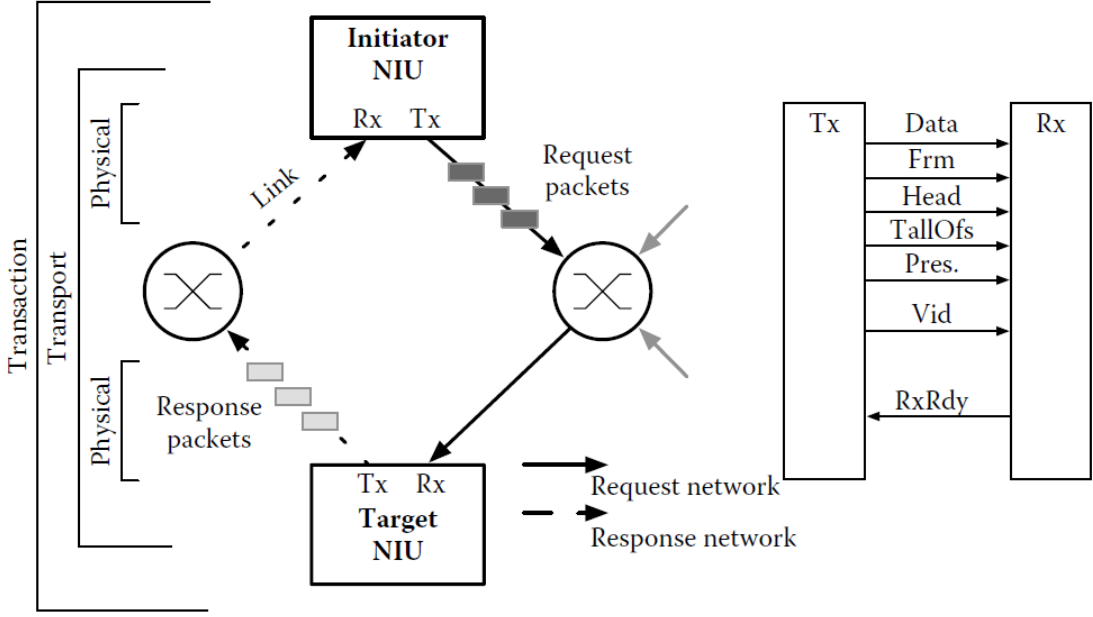
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 735 1808 808">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1877 927">As a further example, “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.1 Initiator NIU Units</p> <p>Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

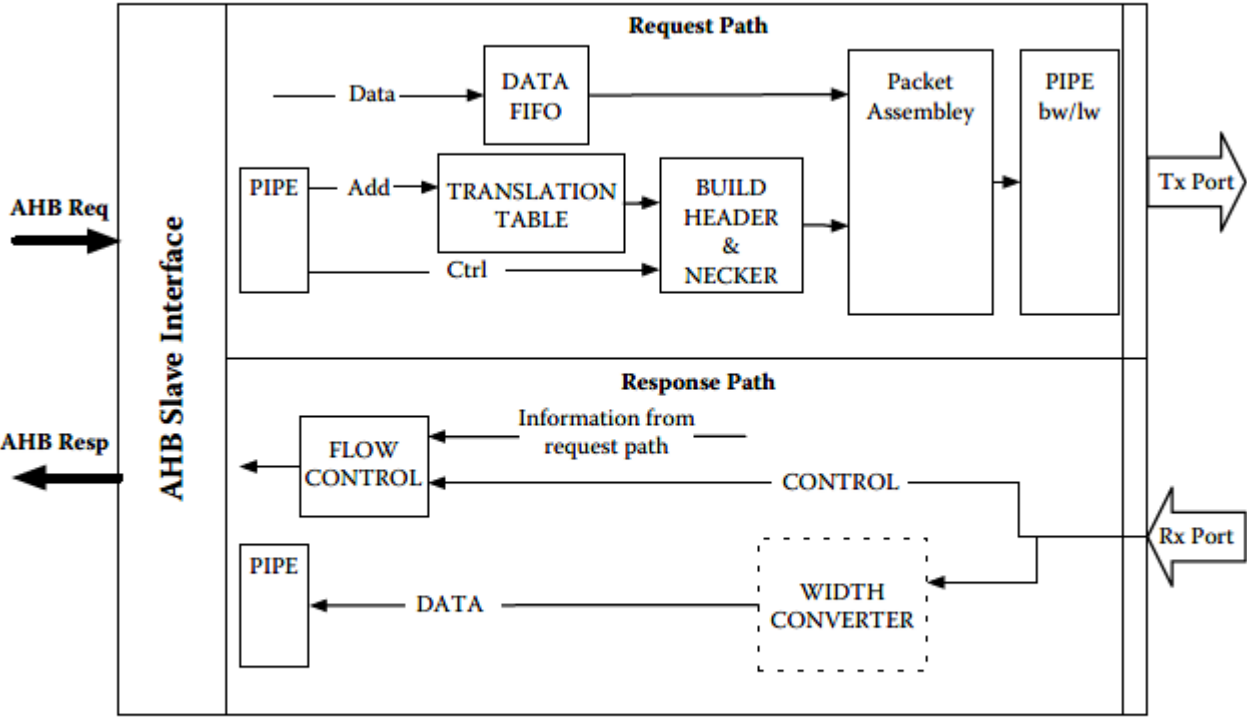
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p>  <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p>
connecting the master interface unit to the	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product connects the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
interconnect so that the master interface unit is between the master and the interconnect;	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

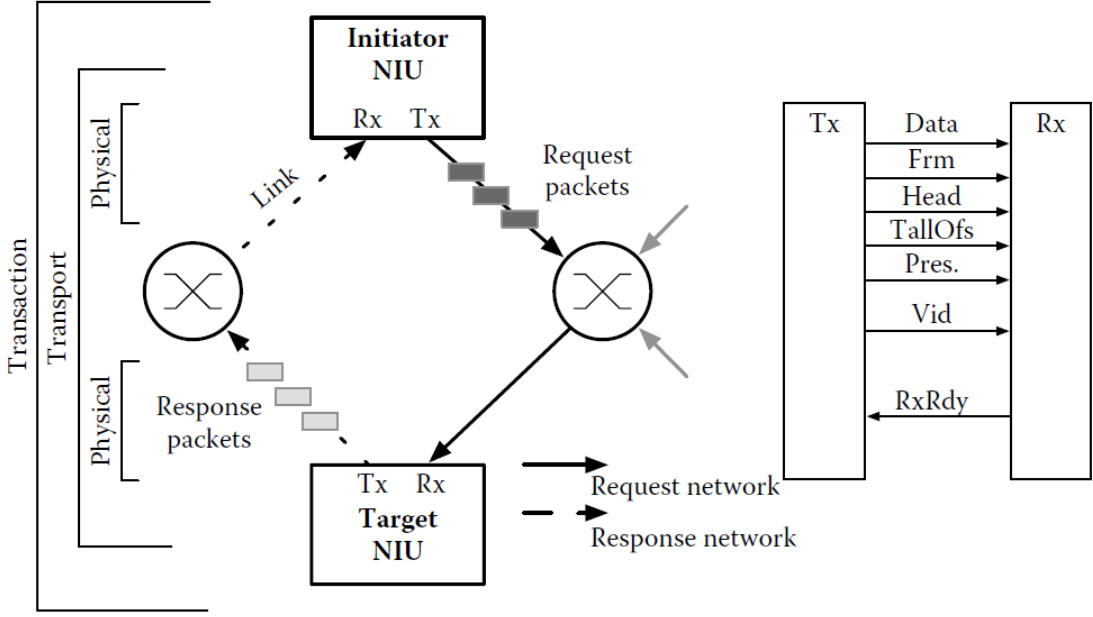
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2 Network Interface Units</p> <p>The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p>As a further example, “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.1 Initiator NIU Units</p> <p>Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

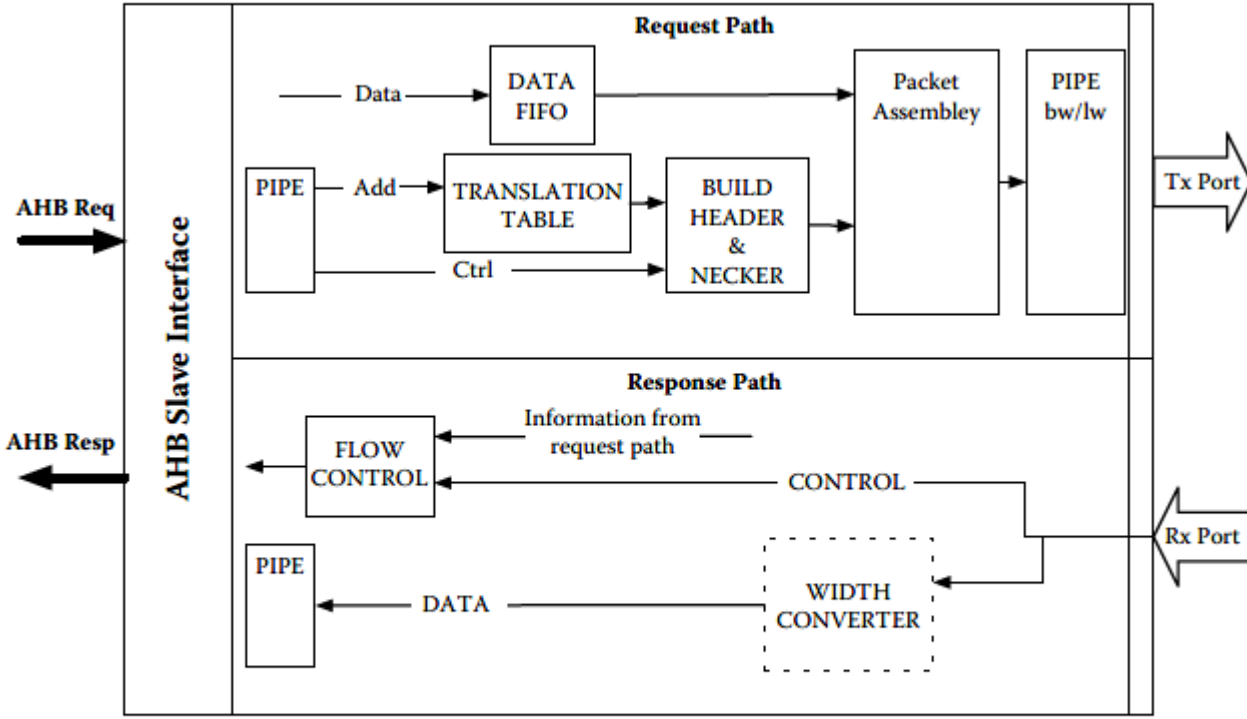
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p>  <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p>
connecting the slave to a slave	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product connecting the slave to a slave interface unit of the interface units, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
interface unit of the interface units;	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

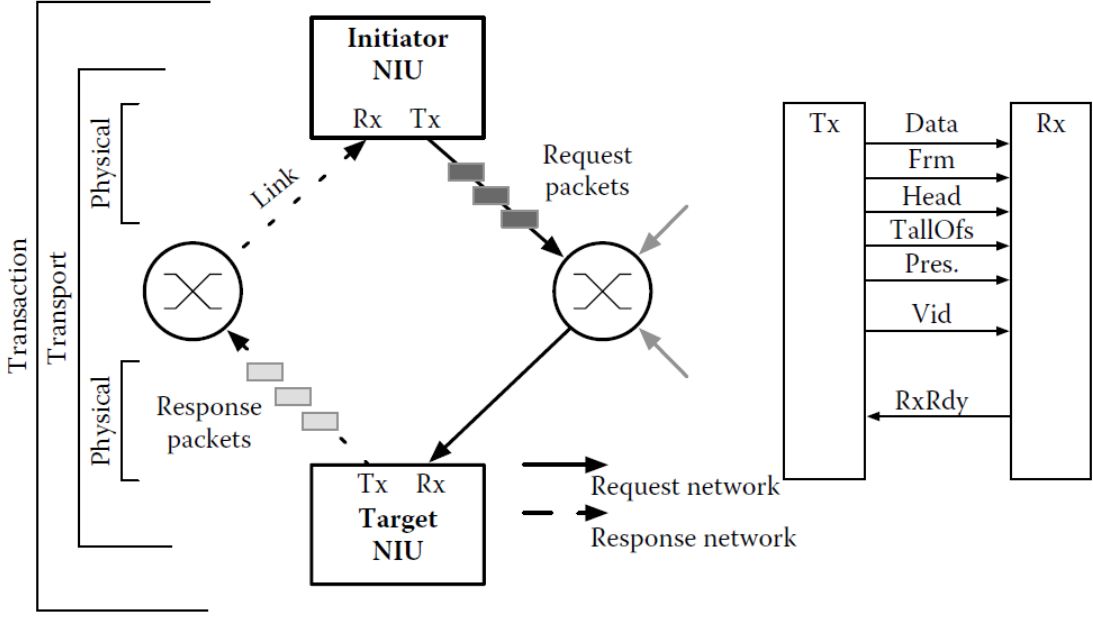
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2 Network Interface Units</p> <p>The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p>As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets”:</p>

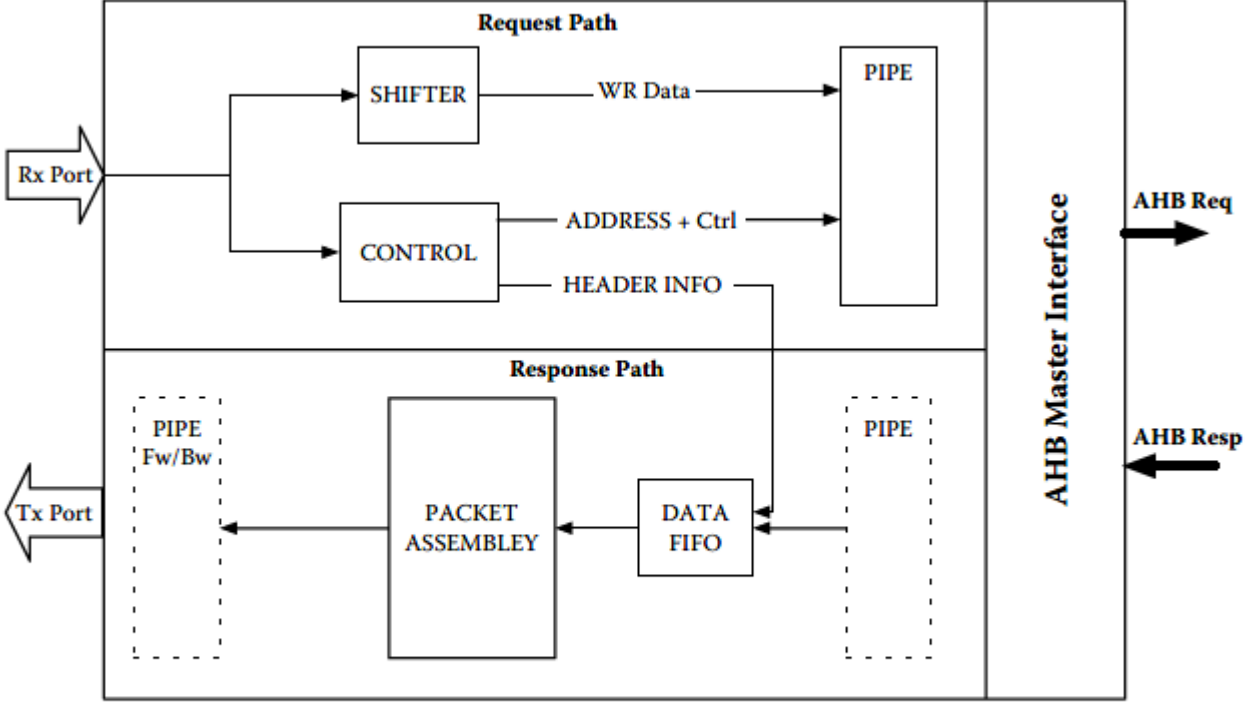
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">Target NIU Architecture</p>  <p>The diagram illustrates the Target NIU Architecture. It is divided into two main horizontal sections: the Request Path (top) and the Response Path (bottom). On the left, there is an Rx Port (input) and a Tx Port (output). The Request Path contains a SHIFTER and a CONTROL block. The Rx Port feeds into both. The SHIFTER outputs WR Data to a PIPE. The CONTROL block outputs ADDRESS + Ctrl to the same PIPE and HEADER INFO to a DATA FIFO in the Response Path. The Response Path contains a PACKET ASSEMBLY, a DATA FIFO, and two dashed boxes labeled PIPE Fw/Bw. The DATA FIFO receives data from the PIPE in the Request Path and outputs to the PACKET ASSEMBLY. The PACKET ASSEMBLY outputs to the Tx Port. The Tx Port also feeds into the left PIPE Fw/Bw block. On the right, the AHB Master Interface connects to the system bus, with AHB Req (request) and AHB Resp (response) signals.</p> <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect;	<p>The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product connects the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect, either literally or under the doctrine of equivalents.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

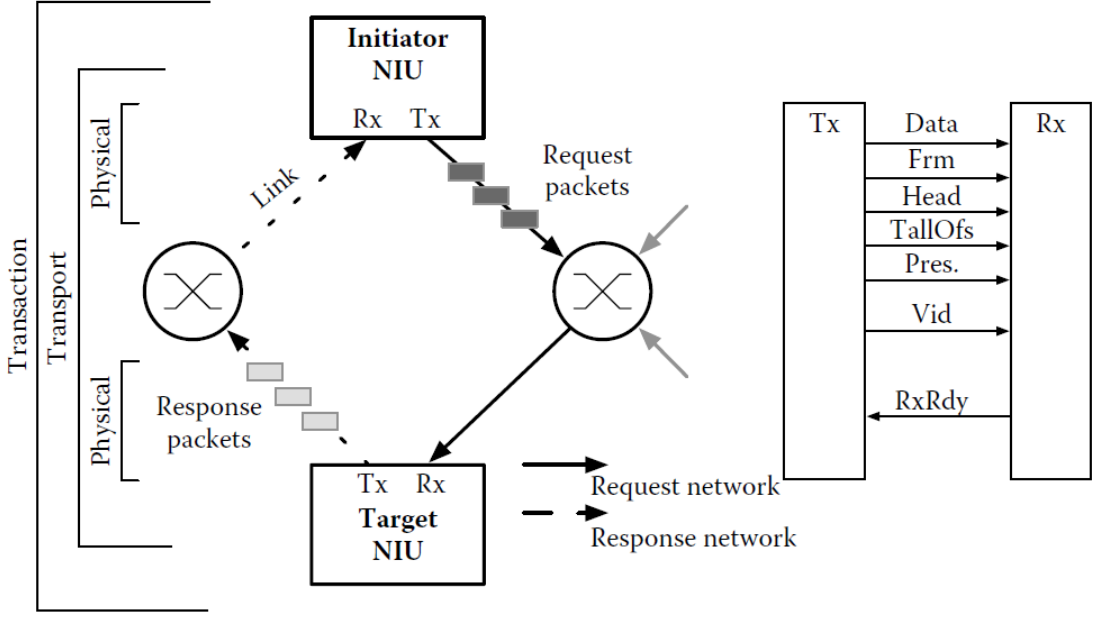
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 298">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 508 1808 704" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 735 1808 808">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1850 967">As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets”:</p>

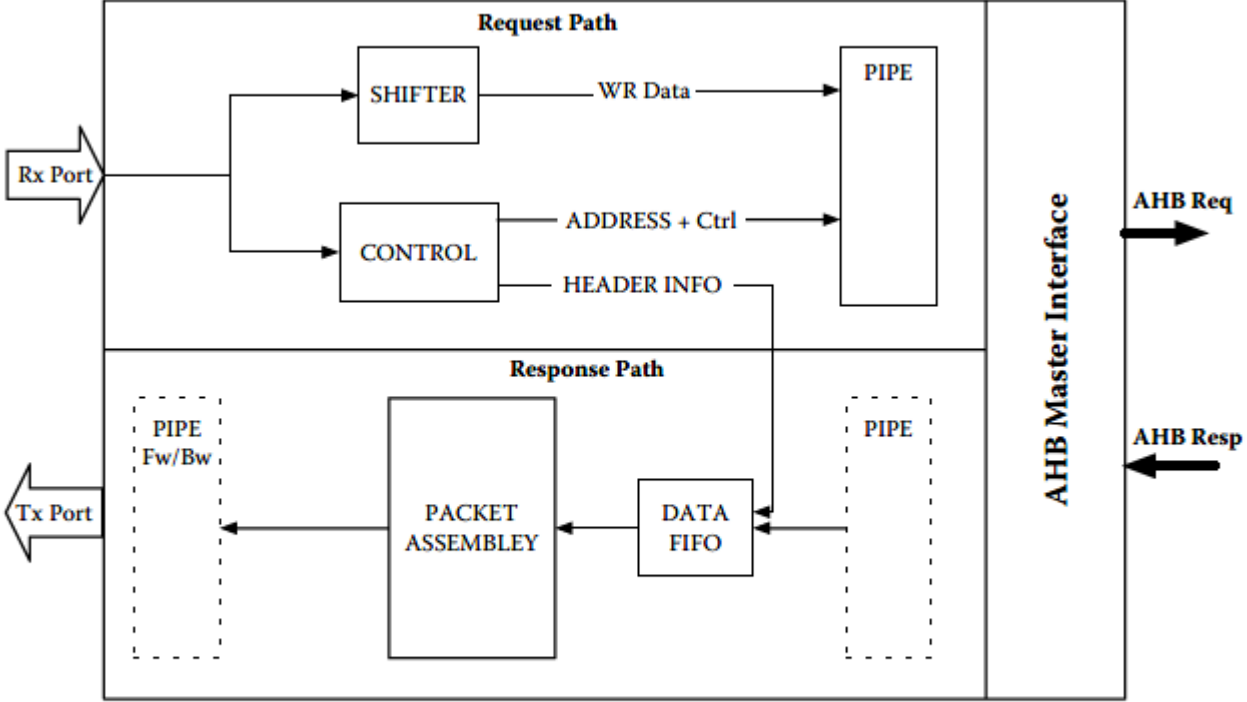
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">Target NIU Architecture</p>  <p>The diagram illustrates the Target NIU Architecture. It is divided into two main horizontal sections: the Request Path (top) and the Response Path (bottom). On the left, there is an Rx Port (input) and a Tx Port (output). The Request Path includes a SHIFTER and a CONTROL block. The Rx Port feeds into both. The SHIFTER outputs WR Data to a PIPE. The CONTROL block outputs ADDRESS + Ctrl to the same PIPE and HEADER INFO to a DATA FIFO in the Response Path. The Response Path includes a PACKET ASSEMBLY and two dashed PIPE blocks labeled PIPE Fw/Bw. The DATA FIFO feeds into the PACKET ASSEMBLY, which then feeds into the Tx Port. On the right, the AHB Master Interface connects to the system bus, with AHB Req (request) and AHB Resp (response) signals.</p> <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>
determining by a master	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product determines by a master determination unit of the master interface unit a first optimal amount of data to be

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit;	<p>buffered by a master wrapper of the master interface unit, either literally or under the doctrine of equivalents.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

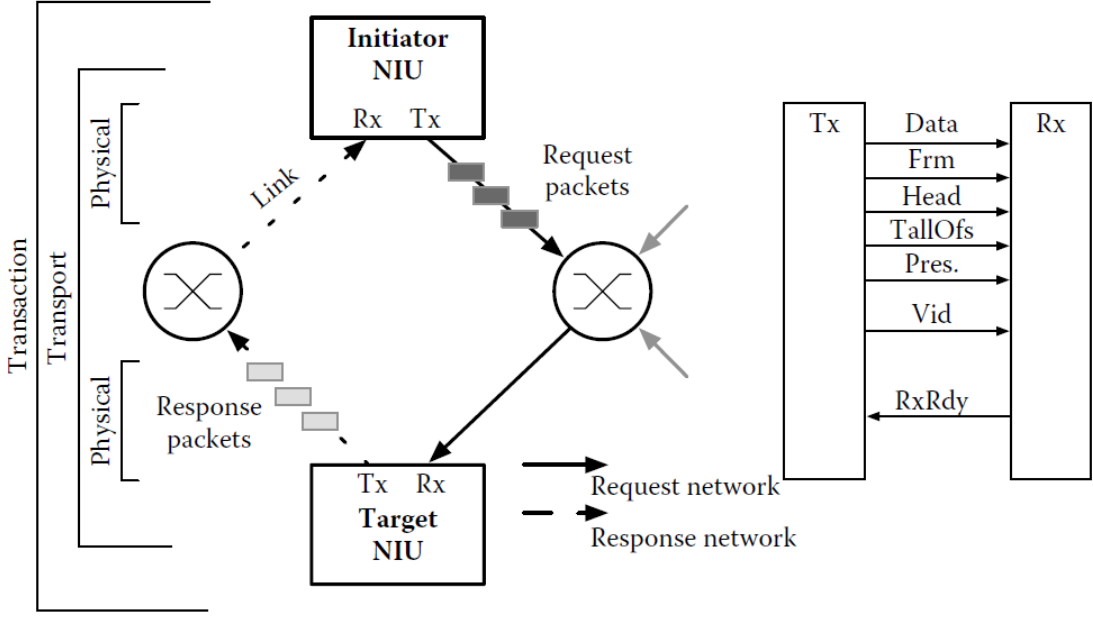
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 732 1808 805">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1835 967">In the Arteris NoC “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC” and includes blocks such as “Data FIFO,” “Translation Table,” “Build Header & Necker,” and “Packet Assembly”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div><p style="text-align: center;">NIU Architecture</p><p>The diagram illustrates the NIU Architecture, which is divided into two main sections: the Request Path and the Response Path. On the left, an 'AHB Slave Interface' is shown with an 'AHB Req' arrow pointing into the Request Path and an 'AHB Resp' arrow pointing out from the Response Path. The Request Path starts with 'Data' entering a 'DATA FIFO', which then feeds into a 'Packet Assembly' block. A 'PIPE' block also feeds into the 'Packet Assembly' via an 'Add' connection. The 'Packet Assembly' outputs to a 'PIPE bw/lw' block, which finally connects to the 'Tx Port'. The Response Path starts at the 'Rx Port', which feeds into a 'WIDTH CONVERTER' (indicated by a dashed box). The 'WIDTH CONVERTER' outputs 'DATA' to a 'PIPE' block. This 'PIPE' feeds into a 'FLOW CONTROL' block. The 'FLOW CONTROL' block has a 'CONTROL' output that feeds back into the 'Packet Assembly' in the Request Path. Additionally, the 'FLOW CONTROL' block receives 'Information from request path' and has an output that feeds back into the 'DATA FIFO'.</p></div> <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317.</p> <p>In the Initiator NIUs of the Arteris NoC, a “FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>defined by the hardware parameter” which “indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port”:</p> <p>11.3.2.1 Initiator NIU Units</p> <p>Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

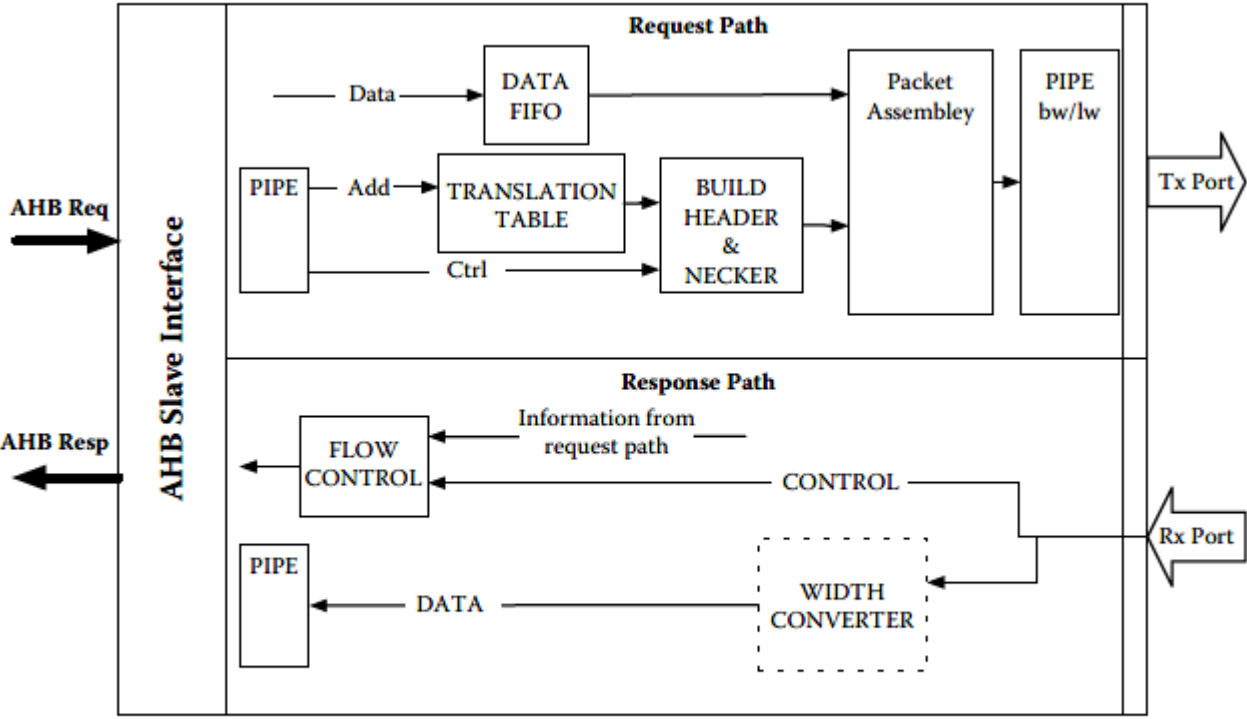
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p>  <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p>
determining by a slave determination unit	The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product determines by a slave determination unit of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit, either literally or under the doctrine of equivalents.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
<p>of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit;</p>	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

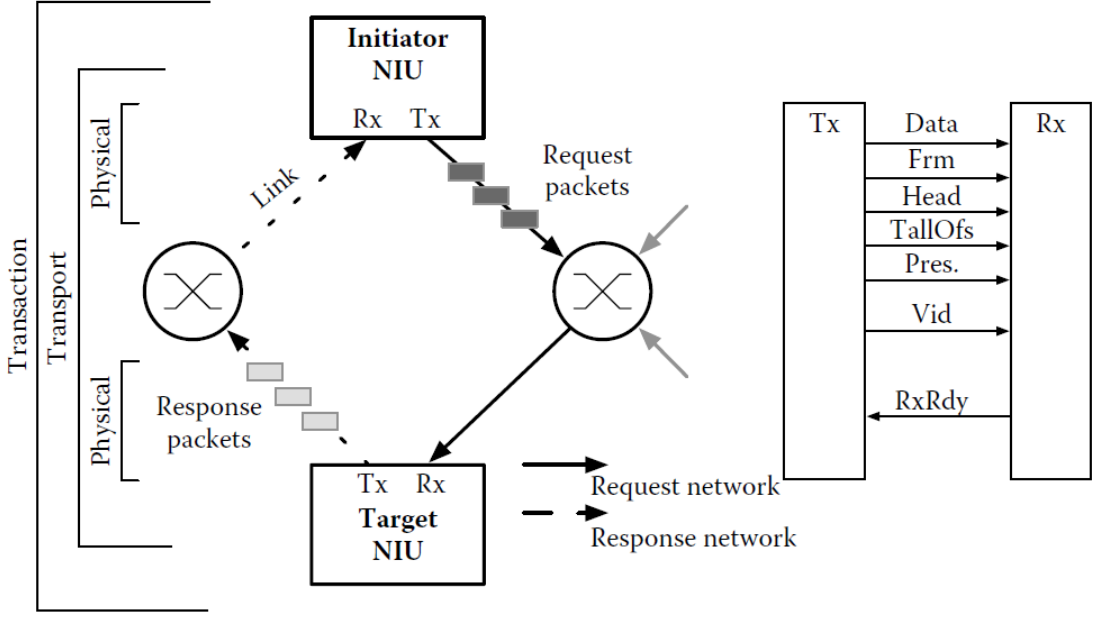
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 732 1808 805">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1871 967">As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets” and includes blocks such as “Data FIFO ”and “Packet Assembly”:</p>

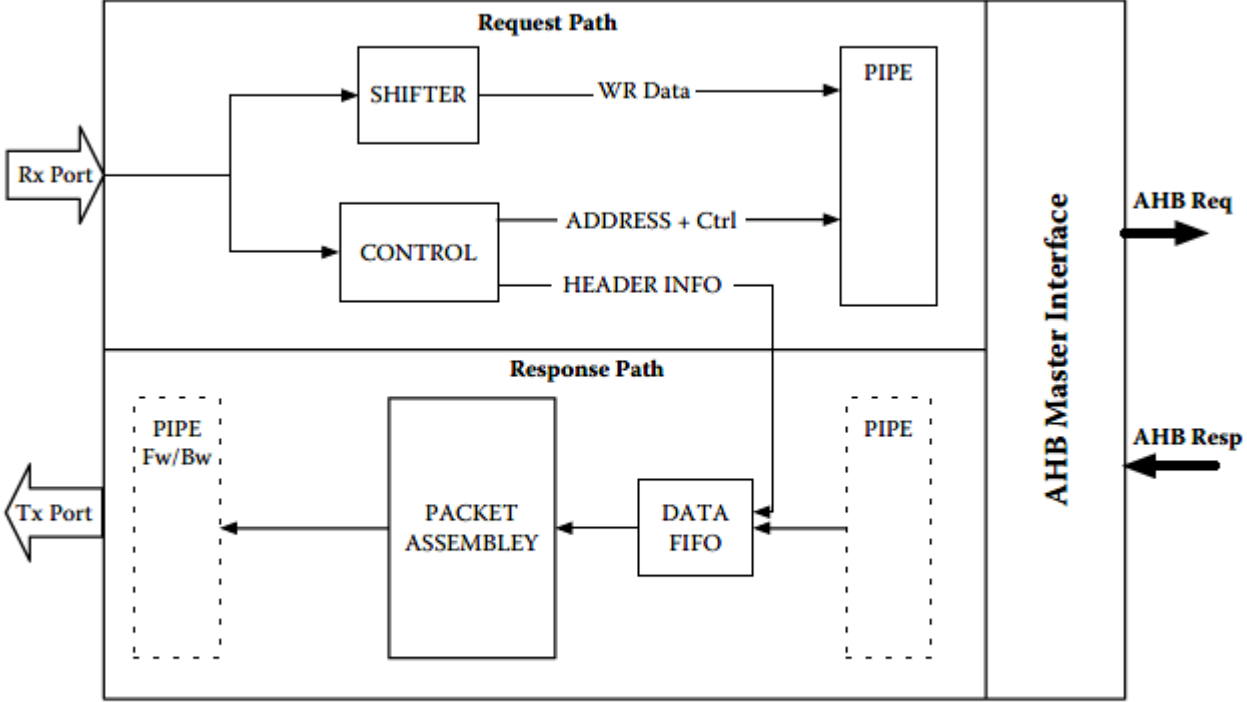
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">Target NIU Architecture</p>  <p>The diagram illustrates the Target NIU Architecture. It is divided into two main horizontal sections: the Request Path (top) and the Response Path (bottom). On the left, there is an Rx Port (input) and a Tx Port (output). The Request Path contains a SHIFTER and a CONTROL block. The Rx Port feeds into both. The SHIFTER outputs WR Data to a PIPE. The CONTROL block outputs ADDRESS + Ctrl to the same PIPE and HEADER INFO to a DATA FIFO in the Response Path. The Response Path contains a PACKET ASSEMBLY, a DATA FIFO, and two dashed boxes labeled PIPE Fw/Bw. The DATA FIFO receives data from the PIPE in the Request Path and outputs to the PACKET ASSEMBLY. The PACKET ASSEMBLY outputs to the Tx Port. On the right, an AHB Master Interface connects the internal components to the AHB bus, with AHB Req (request) and AHB Resp (response) signals.</p> <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, “[a] FIFO memory is inserted in the datapath for AHB ... accesses. The FIFO memory absorbs data at the AHB ... rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is defined by the hardware parameter” which “indicates the amount of data required to generate a ... packet: each time the FIFO is full, a ... packet is sent on the Tx port”:</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0 , at 317-318.
buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered;	<p>The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product buffers by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered, either literally or under the doctrine of equivalents.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

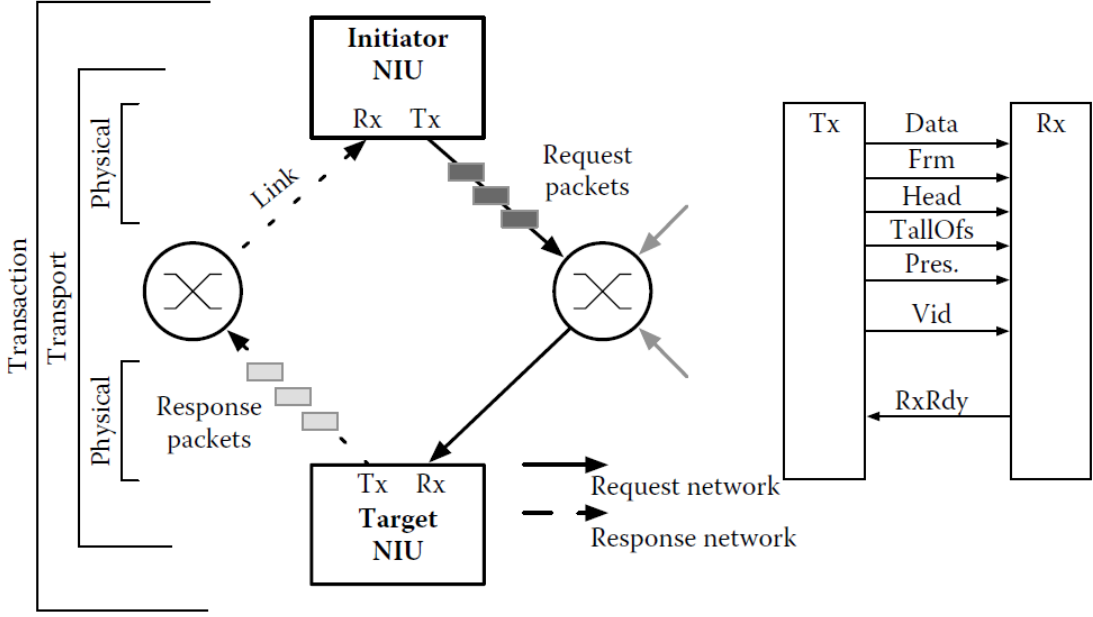
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 735 1808 808">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1871 967">As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets” and includes blocks such as “Data FIFO ”and “Packet Assembly”:</p>

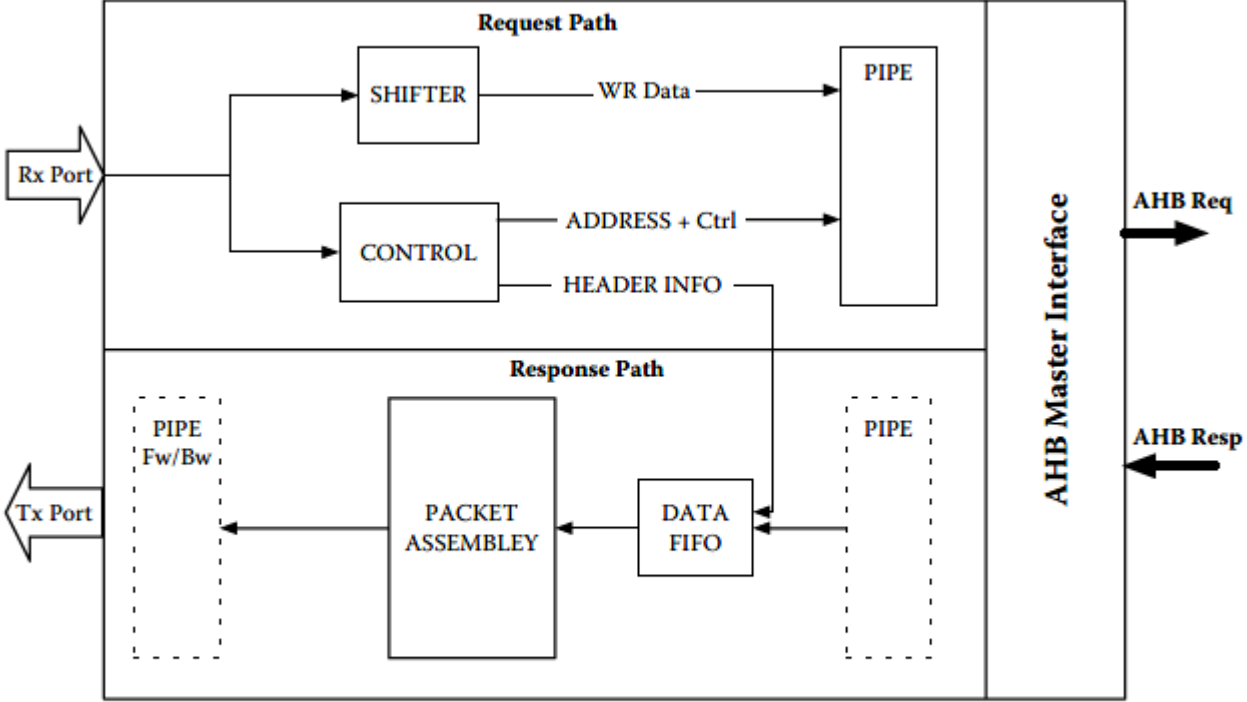
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">Target NIU Architecture</p>  <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, “[a] FIFO memory is inserted in the datapath for AHB ... accesses. The FIFO memory absorbs data at the AHB ... rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is defined by the hardware parameter” which “indicates the amount of data required to generate a ... packet: each time the FIFO is full, a ... packet is sent on the Tx port”:</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p> <p>As a further illustration, the Arteris NoC uses “a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency.” For other traffic, the “[b]est effort traffic can be left untouched[,]” “[l]atency sensitive traffic may have its urgency modulated as a function of the transaction[,]” “[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]” and “[o]n the real-time modem data port, the hurry is fixed at a critical level”:</p> <p>Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency.</p> <p>In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links.</p> <p>See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16.</p>
transferring the buffered data from the slave wrapper to the master	<p>The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product transfers the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper, either literally or under the doctrine of equivalents.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
<p>wrapper when said first optimal amount of data has been buffered by the slave wrapper;</p>	<p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

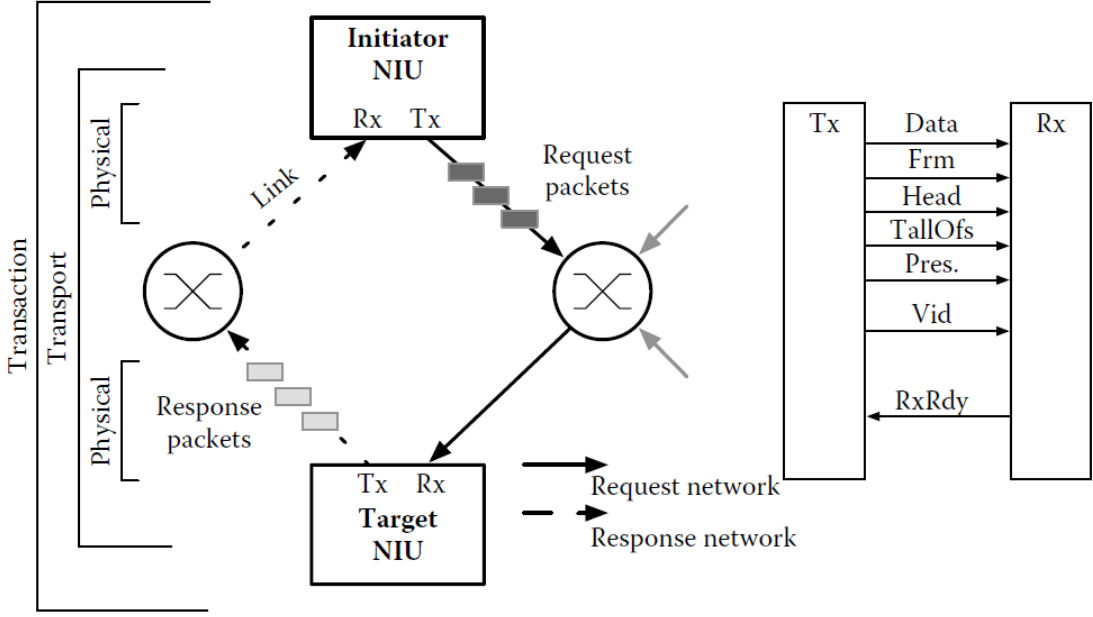
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 735 1808 808">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1871 967">As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets” and includes blocks such as “Data FIFO ”and “Packet Assembly”:</p>

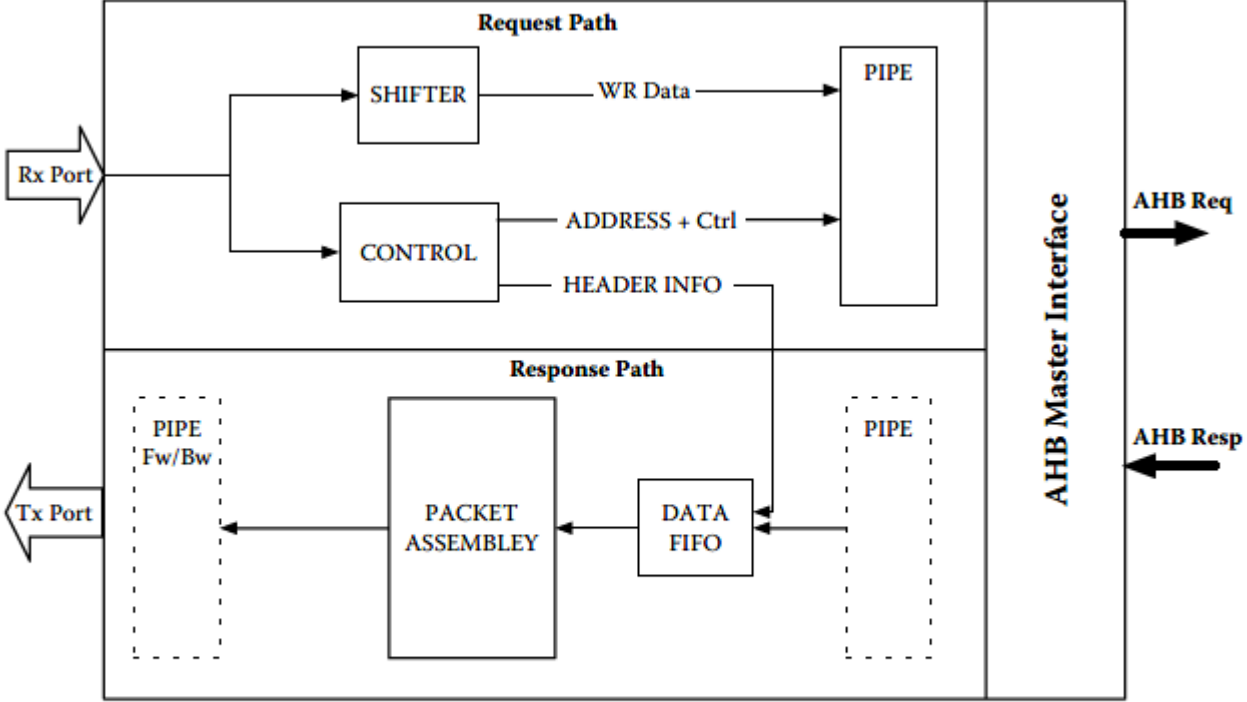
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">Target NIU Architecture</p>  <p>The diagram illustrates the Target NIU Architecture. It is divided into two main horizontal sections: the Request Path (top) and the Response Path (bottom). On the left, an Rx Port (indicated by a right-pointing arrow) feeds into the Request Path. The Request Path contains a SHIFTER and a CONTROL block. The SHIFTER outputs WR Data to a PIPE. The CONTROL block outputs ADDRESS + Ctrl to the same PIPE and HEADER INFO to a DATA FIFO in the Response Path. The Response Path contains a PACKET ASSEMBLY and a DATA FIFO. The DATA FIFO receives data from the PIPE in the Request Path and outputs to the PACKET ASSEMBLY. The PACKET ASSEMBLY outputs to a PIPE Fw/Bw (indicated by a left-pointing arrow) and a Tx Port (indicated by a left-pointing arrow). On the right, an AHB Master Interface connects the internal components to the AHB bus. It shows AHB Req (outgoing) and AHB Resp (incoming) signals.</p> <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, “[a] FIFO memory is inserted in the datapath for AHB ... accesses. The FIFO memory absorbs data at the AHB ... rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is defined by the hardware parameter” which “indicates the amount of data required to generate a ... packet: each time the FIFO is full, a ... packet is sent on the Tx port”:</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0 , at 317-318.
buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper;	<p>The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product buffers by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper, either literally or under the doctrine of equivalents.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

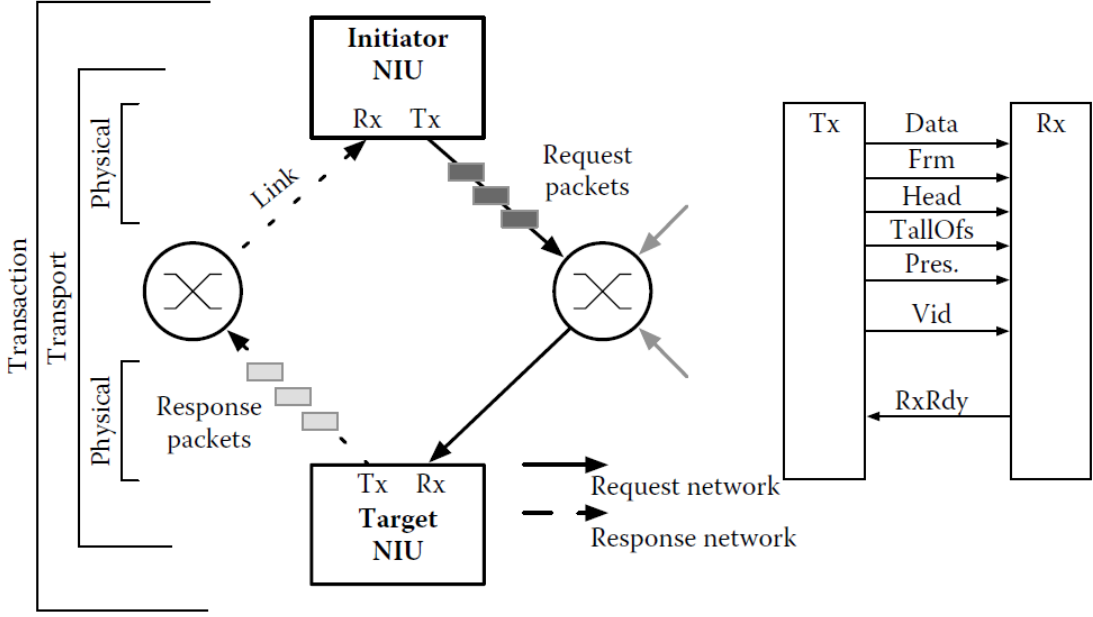
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC”:</p>

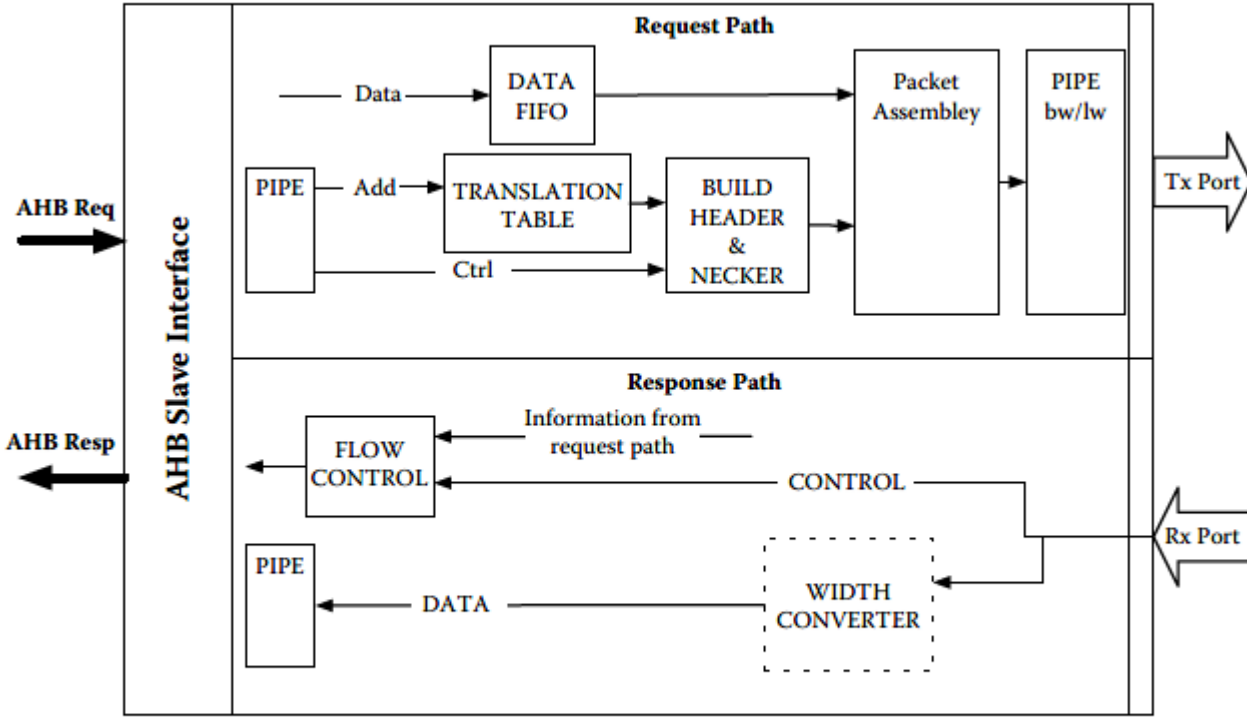
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="520 261 1094 302">11.3.2 Network Interface Units</p> <p data-bbox="520 334 1856 472">The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul data-bbox="594 513 1808 708" style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p data-bbox="499 732 1808 805">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p data-bbox="499 854 1835 967">In the Arteris NoC “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC” and includes blocks such as “Data FIFO,” “Translation Table,” “Build Header & Necker,” and “Packet Assembly”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p>  <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317.</p> <p>In the Initiator NIUs of the Arteris NoC, a “FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>defined by the hardware parameter” which “indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port”:</p> <p>11.3.2.1 Initiator NIU Units</p> <p>Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div><p style="text-align: center;">NIU Architecture</p><p>The diagram illustrates the NIU Architecture, which is divided into two main sections: the Request Path and the Response Path. On the left, an 'AHB Slave Interface' is shown with an 'AHB Req' arrow pointing into the Request Path and an 'AHB Resp' arrow pointing out from the Response Path. The Request Path starts with 'Data' entering a 'DATA FIFO', which then feeds into a 'Packet Assembly' block. A 'PIPE' block also feeds into the 'Packet Assembly' via an 'Add' signal. The 'Packet Assembly' outputs to a 'PIPE bw/lw' block, which then connects to the 'Tx Port'. The Response Path starts at the 'Rx Port', which feeds into a 'WIDTH CONVERTER' (indicated by a dashed box). The 'WIDTH CONVERTER' outputs 'DATA' to a 'PIPE' block. This 'PIPE' feeds into a 'FLOW CONTROL' block. The 'FLOW CONTROL' block has a 'CONTROL' signal that feeds back into the 'Packet Assembly' in the Request Path. Additionally, the 'FLOW CONTROL' block receives 'Information from request path' and outputs a 'Ctrl' signal to a 'BUILD HEADER & NECKER' block. This block also receives an 'Add' signal from a 'PIPE' block and feeds into a 'TRANSLATION TABLE'. The 'TRANSLATION TABLE' outputs to the 'BUILD HEADER & NECKER' block, which then feeds into the 'Packet Assembly'.</p></div> <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p> <p>As a further illustration, the Arteris NoC uses “a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency.” For other traffic, the “[b]est effort traffic can be left untouched[,]” “[l]atency sensitive traffic may have its</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>urgency modulated as a function of the transaction[,]” “[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]” and “[o]n the real-time modem data port, the hurry is fixed at a critical level”:</p> <p>Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency.</p> <p>In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links.</p> <p>See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16.</p>
transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,	<p>The Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product transfers the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper, either literally or under the doctrine of equivalents.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p>

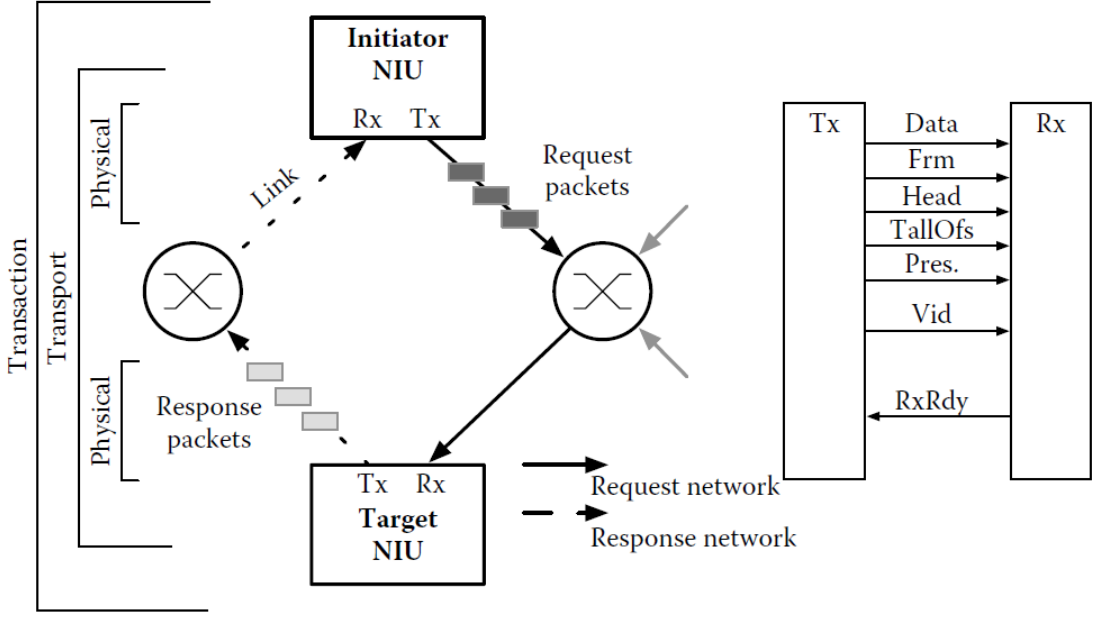
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="562 266 1020 305">11.3.1.1 Transaction Layer</p> <p data-bbox="562 323 1822 500">The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul data-bbox="632 545 1350 639" style="list-style-type: none"> <li data-bbox="632 545 1199 581">• A master sends request packets. <li data-bbox="632 599 1350 639">• Then, the slave returns response packets. <p data-bbox="562 685 1843 1256">As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC”:</p>

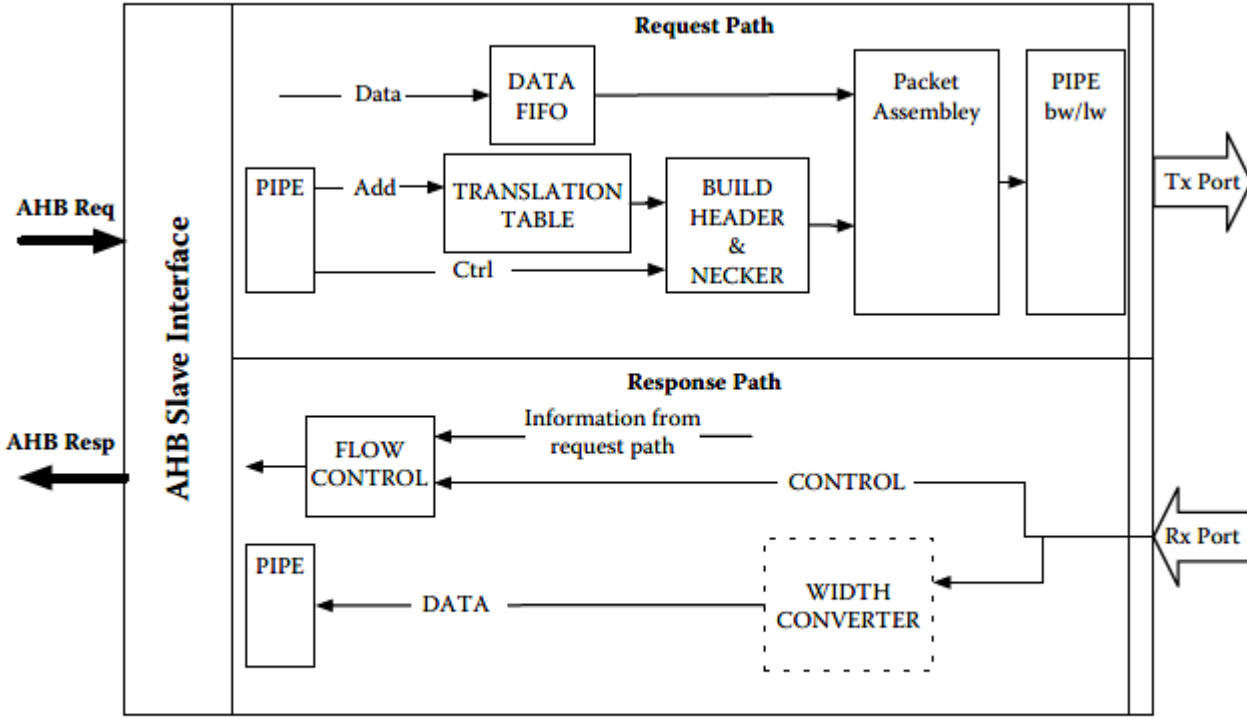
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2 Network Interface Units</p> <p>The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p>In the Arteris NoC “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC” and includes blocks such as “Data FIFO,” “Translation Table,” “Build Header & Necker,” and “Packet Assembly”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p>  <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317.</p> <p>In the Initiator NIUs of the Arteris NoC, a “FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="506 250 1860 326">defined by the hardware parameter” which “indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port”:</p> <p data-bbox="531 386 1024 423">11.3.2.1 Initiator NIU Units</p> <p data-bbox="527 448 1812 1081">Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p data-bbox="527 1089 1812 1170">A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

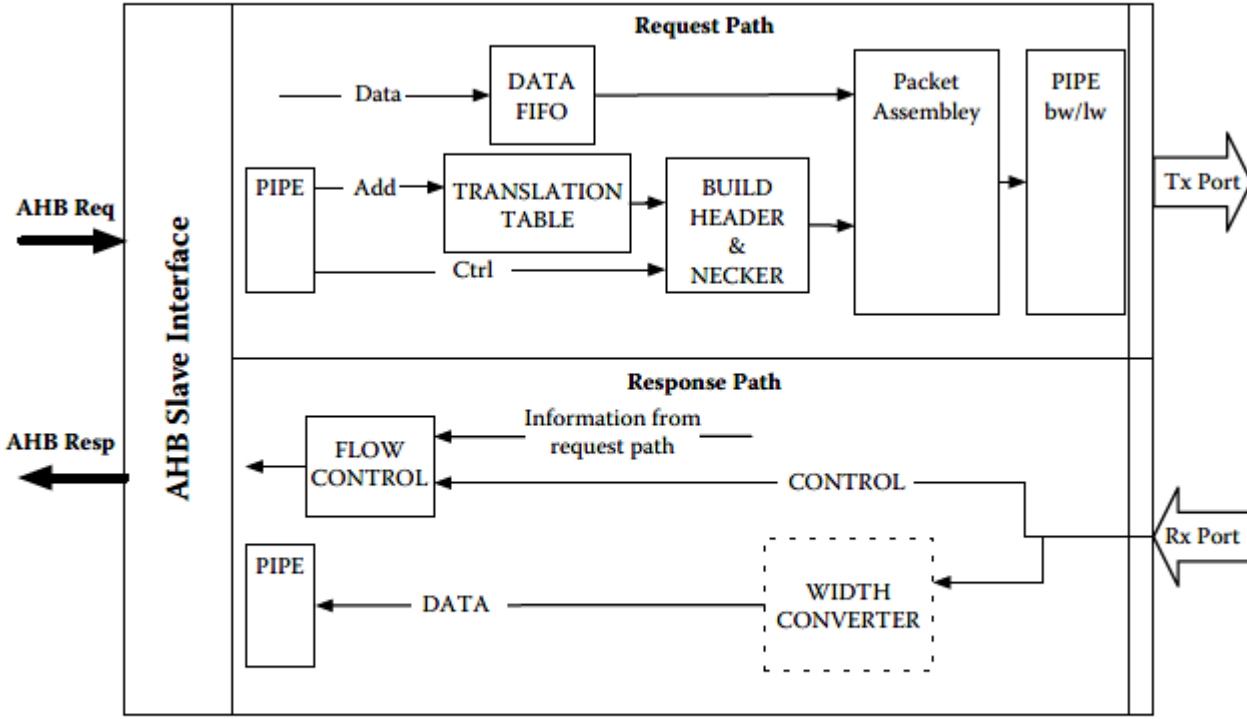
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p style="text-align: center;">NIU Architecture</p>  <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p>
wherein at least one of the first determination unit	In the Arteris NoC utilized by the Snapdragon SoC included in the Lenovo product, at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
<p>and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it</p>	<p>to communication properties of the communication between the master and the slave wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery, either literally or under the doctrine of equivalents.</p> <p>For example, the Arteris NoC uses Network Interface Units (NIUs), which “translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols” and in the Arteris NoC, the NIUs “are at the boundary of the NoC” and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:</p> <p>11.3.1.1 Transaction Layer</p> <p>The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers:</p> <ul style="list-style-type: none"> • A master sends request packets. • Then, the slave returns response packets. <p>As shown in Figure 11.1, requests from an initiator are sent through the master NIU’s transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets</p>

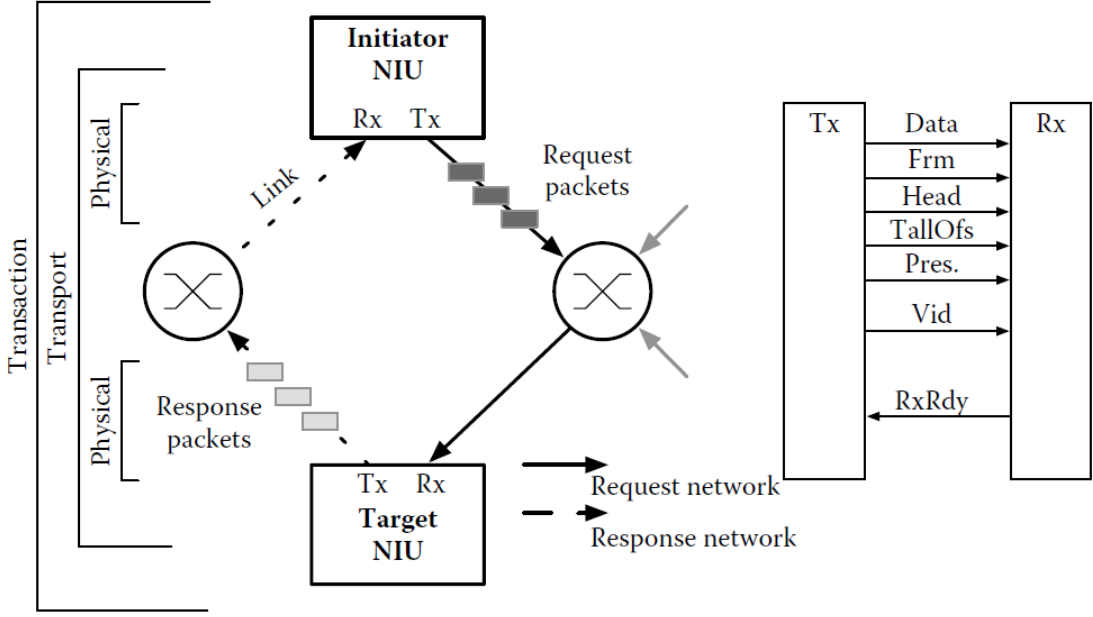
U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
<p>is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.</p>	<p>on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	 <p>FIGURE 11.1 NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.</p> <p>See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.</p> <p>The Initiator NIUs are “used to connect a master node to the NoC” and the Target NIUs are “used to connect a slave node to the NoC”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2 Network Interface Units</p> <p>The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated:</p> <ul style="list-style-type: none"> • Initiator NIU—third party protocol-to-NTTP, used to connect a master node to the NoC • Target NIUs—NTTP-to-third party protocol, used to connect a slave node to the NoC <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317.</p> <p>In the Arteris NoC “Initiator NIU units...enable connection between an AMBA-AHB master IP and the NoC” and includes blocks such as “Data FIFO,” “Translation Table,” “Build Header & Necker,” and “Packet Assembly”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div><p style="text-align: center;">NIU Architecture</p><p>The diagram illustrates the NIU Architecture, which is divided into two main sections: the Request Path and the Response Path. On the left, an 'AHB Slave Interface' is shown with an 'AHB Req' arrow pointing into the Request Path and an 'AHB Resp' arrow pointing out from the Response Path. The Request Path starts with 'Data' entering a 'DATA FIFO', which then feeds into a 'Packet Assembly' block. A 'PIPE' block also feeds into the 'Packet Assembly' via an 'Add' connection. The 'Packet Assembly' outputs to a 'PIPE bw/lw' block, which then connects to the 'Tx Port'. The Response Path starts at the 'Rx Port', which feeds into a 'WIDTH CONVERTER' (indicated by a dashed box). The 'WIDTH CONVERTER' outputs 'DATA' to a 'PIPE' block. This 'PIPE' block feeds into a 'FLOW CONTROL' block. The 'FLOW CONTROL' block has a 'CONTROL' output that feeds back into the 'Packet Assembly' in the Request Path. Additionally, the 'FLOW CONTROL' block receives 'Information from request path' and has an output that feeds back into the 'DATA FIFO'.</p></div> <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317.</p> <p>In the Initiator NIUs of the Arteris NoC, a “FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>defined by the hardware parameter” which “indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port”:</p> <p>11.3.2.1 Initiator NIU Units</p> <p>Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU.</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div><p style="text-align: center;">NIU Architecture</p><p>The diagram illustrates the NIU Architecture, which is divided into two main sections: the Request Path and the Response Path. On the left, an 'AHB Slave Interface' is shown with an 'AHB Req' arrow pointing into the Request Path and an 'AHB Resp' arrow pointing out from the Response Path. The Request Path starts with 'Data' entering a 'DATA FIFO', which then feeds into a 'Packet Assembly' block. A 'PIPE' block also feeds into the 'Packet Assembly' via an 'Add' signal. The 'Packet Assembly' outputs to a 'PIPE bw/lw' block, which finally connects to the 'Tx Port'. The Response Path starts at the 'Rx Port', which feeds into a 'WIDTH CONVERTER' (indicated by a dashed box). The 'WIDTH CONVERTER' outputs 'DATA' to a 'PIPE' block. This 'PIPE' feeds into a 'FLOW CONTROL' block. The 'FLOW CONTROL' block has a 'CONTROL' signal that feeds back into the 'Packet Assembly' in the Request Path. Additionally, the 'FLOW CONTROL' block receives 'Information from request path' and outputs an 'Add' signal to the 'TRANSLATION TABLE'. The 'TRANSLATION TABLE' also receives a 'Ctrl' signal from the 'FLOW CONTROL' block and outputs to the 'BUILD HEADER & NECKER' block. The 'BUILD HEADER & NECKER' block outputs to the 'Packet Assembly'.</p></div> <p>FIGURE 11.4 Network interface unit: Initiator architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p> <p>As further example, “Target NIU units enable connection of a slave IP to the NoC by translating NTPP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets” and includes blocks such as “Data FIFO ”and “Packet Assembly”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>11.3.2.2 Target NIU Units</p> <p>Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<div><p>Target NIU Architecture</p><p>The diagram illustrates the Target NIU Architecture. It is divided into two main horizontal sections: the Request Path (top) and the Response Path (bottom). The Request Path starts with an Rx Port (indicated by a right-pointing arrow) that splits into two parallel paths. The upper path goes through a SHIFTER block, which outputs WR Data to a PIPE block. The lower path goes through a CONTROL block, which outputs ADDRESS + Ctrl and HEADER INFO to the same PIPE block. The Response Path starts with an AHB Master Interface on the right, which receives AHB Resp (indicated by a left-pointing arrow) and sends AHB Req (indicated by a right-pointing arrow). This interface connects to a DATA FIFO block. The DATA FIFO outputs to a PACKET ASSEMBLY block, which then outputs to a PIPE Fw/Bw block (indicated by a dashed box). The Tx Port (indicated by a left-pointing arrow) is connected to the output of the PACKET ASSEMBLY block.</p></div> <p>FIGURE 11.5 Network interface unit: Target architecture.</p> <p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, “[a] FIFO memory is inserted in the datapath for AHB ... accesses. The FIFO memory absorbs data at the AHB ... rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received.” “[T]he FIFO depth is defined by the hardware parameter” which “indicates the amount of data required to generate a ... packet: each time the FIFO is full, a ... packet is sent on the Tx port”:</p> <p>A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is</p> <ul style="list-style-type: none"> • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</p> <p>As a further illustration, the “Arteris NTTP protocol is packet-based” and the packets, which have “header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address,” are “transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes”:</p> <p>11.3.1.2 Transport Layer</p> <p>The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target.</p> <p><i>Id.</i> at 313.</p> <p>As yet a further illustration, packets in the Arteris NoC are “delivered as words that are sent along links and “[o]ne link (represented in Figure 11.1) defines the following signals,” which include “the current priority of the packet used to define preferred traffic class (or Quality of Service)” and “[f]low control”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<ul style="list-style-type: none"> • Data—Data word of the width specified at design-time. • Frm—When asserted high, indicates that a packet is being transmitted. • Head—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only. • TailOfs—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only. • Pres.—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2). • Vld—Data valid: when asserted high, indicates that a word is being transmitted. • RxRdy—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy. <p><i>Id.</i> at 313-314.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>As a further illustration, the Arteris NoC implements Quality of Service (QoS) to “provide[] a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic”; “QoS, which includes guarantees of throughput and/or latency, is achieved by exploiting the signal pressure embedded into the NTTP packet definition” where the “pressure signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed”; and the “pressure information will be embedded in the NTTP packet at the NIU level”:</p> <p>Quality of Service (QoS). The QoS is a very important feature in the inter-connect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in <i>Æthereal NoC</i> [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT.</p> <p>In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair.</p> <p>The Arteris NoC supports the following four different traffic classes:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<ul style="list-style-type: none"> • Real time and low latency (RTLL)—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • Guaranteed throughput (GT)—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • Guaranteed bandwidth (GBW)—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • Best effort (BE)—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p data-bbox="541 282 1843 358">* Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic.</p> <p data-bbox="499 428 1801 505">Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 315-316.</p> <p data-bbox="499 548 1885 727">In addition, the Arteris Interconnect includes “a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency.” For other traffic, the “[b]est effort traffic can be left untouched[,]” “[l]atency sensitive traffic may have its urgency modulated as a function of the transaction[,]” “[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]” and “[o]n the real-time modem data port, the hurry is fixed at a critical level.”</p> <p data-bbox="512 781 1864 1036">Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency.</p> <p data-bbox="512 1045 1864 1170">In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links.</p> <p data-bbox="499 1219 1829 1333">Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springer-appdrivennocarchitecture8.5x11.pdf, at p. 16.</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<p>For the other traffic, the configuration can be done in architecture.</p> <ul style="list-style-type: none"> • Best effort traffic can be left untouched. • Latency sensitive traffic may have its urgency modulated as a function of the transaction: <i>Normal</i> for writes and <i>important</i> for reads. • Soft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives: <i>Critical</i> until a specified bandwidth is obtained on a sliding 4 microsecond window, and <i>normal</i> thereafter. These settings are set through configuration registers and may be modified while the interconnect is running. The mechanism is called a bandwidth regulator. • On the real-time modem data port, the hurry is fixed at a critical level. <p><i>Id.</i> at 18.</p> <p>As a further illustration, the Arteris NoC implements QoS mechanisms that performs arbitration based on “Bandwidth Regularator (BR)” and “Bandwidth Limiter (BL)”:</p>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)

“Integrated circuit and method for buffering to optimize burst length in networks on chips”

'800 Patent Claim	Lenovo Product Including Snapdragon System on Chip ¹
	<h2 data-bbox="527 261 1478 326">Bandwidth Limiters and Rate Regulators</h2> <p data-bbox="527 391 1688 618">Many times architects will want to implement QoS within their SoC but the QoS prioritization data is not available from the individual IP blocks. In this case, QoS information may be generated from within the NoC interconnect using Arteris' QoS Generator. The QoS Generator can instantiate sophisticated, and software programmable, means to regulate interconnect QoS, including:</p> <ul style="list-style-type: none"> <li data-bbox="575 683 1640 805">➤ Bandwidth Limiters – Bandwidth limiters cause a socket to stop accepting requests when a run-time programmable throughput threshold has been exceeded. <li data-bbox="575 824 1703 1000">➤ Rate Regulators – Rate regulators cause a socket's transactions to be demoted when a bandwidth threshold is reached. This can be considered a smoother version of the bandwidth limiter because transactions are only demoted instead of stalled. <p data-bbox="499 1027 1356 1060">https://www.arteris.com/end-to-end-quality-of-service-qos</p>